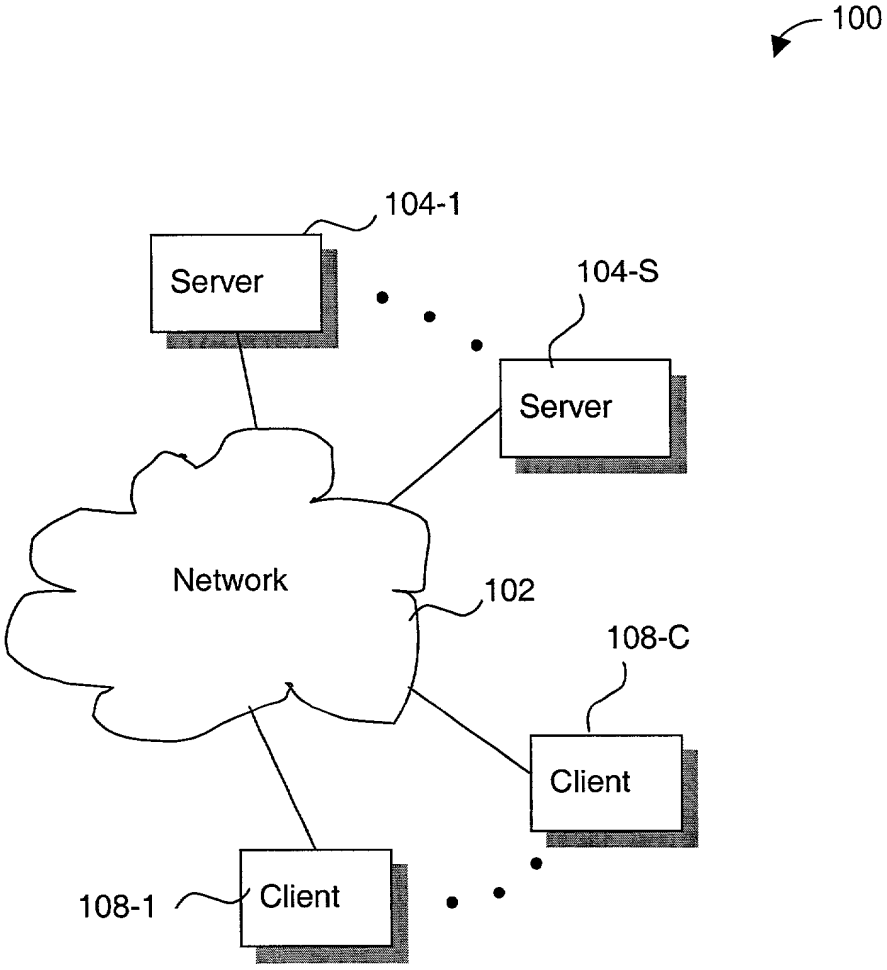

FIG. 1 (Prior Art)

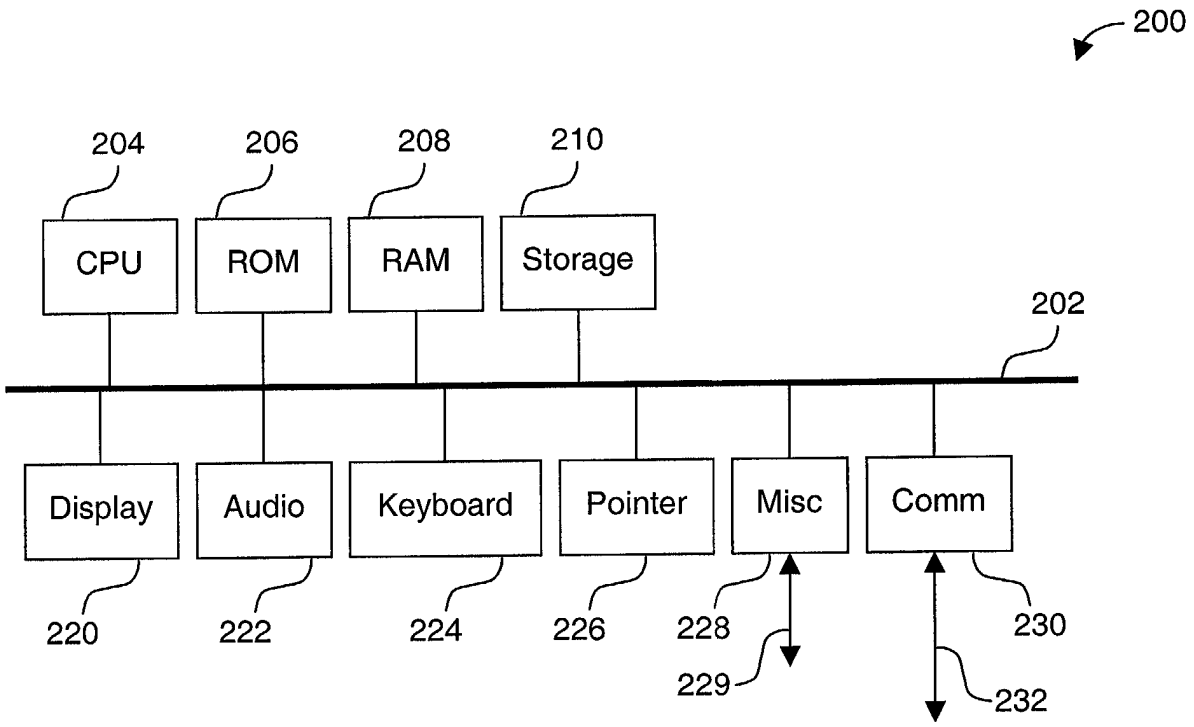

FIG. 2 (Prior Art)

300

Employee 301

Name 302	DOB 304	Fitness 306
Patrick 308	1/1/1	Good
Nancy 310	10/10/10	Bad
Bill 312	1/2/3	Okay

FIG. 3A (Prior Art)

320

Employee 301

Name 302	DOB 304	Fitness 306	Diet 322
Patrick 308	1/1/1	Good	Healthy
Nancy 310	10/10/10	Bad	Poor
Bill 312	1/2/3	Okay	Okay

FIG. 3B (Prior Art)

340

Employee 301

Name 302	DOB 304	Fitness 306	Breakfast 342	Lunch 344	Dinner 346
Patrick 308	1/1/1	Good	Healthy	Okay	Poor
Nancy 310	10/10/10	Bad	Okay	Healthy	Poor
Bill 312	1/2/3	Okay	Okay	Poor	Healthy

FIG. 3C (Prior Art)

400

Employee 301

Name 302	DOB 304	Fitness 306	Diet 422
Patrick 308	1/1/1	Good	T2
Nancy 310	10/10/10	Bad	T2
Bill 312	1/2/3	Okay	T2

FIG. 4A (Prior Art)

440

Diet 423

Name 302	Breakfast 442	Lunch 444	Dinner 446
Patrick 308	Healthy	Okay	Poor
Nancy 310	Okay	Healthy	Poor
Bill 312	Okay	Poor	Healthy

FIG. 4B (Prior Art)

500

Employee 301

Name 302	DOB 304	Fitness 306	Diet 522
Patrick 308	1/1/1	Good	T2
Nancy 310	10/10/10	Bad	T2
Bill 312	1/2/3	Okay	T2

FIG. 5A (Prior Art)

540

Diet 523

Name 302	Breakfast 542	Lunch 544	Dinner 546
Patrick 308	Healthy	Okay	T3
Nancy 310	Okay	Healthy	T3
Bill 312	Okay	Poor	T3

FIG. 5B (Prior Art)

560

Dinner 547

Name 302	Early 562	Late 564
Patrick 308	Healthy	Okay
Nancy 310	Poor	Healthy
Bill 312	Okay	Poor

FIG. 5C (Prior Art)

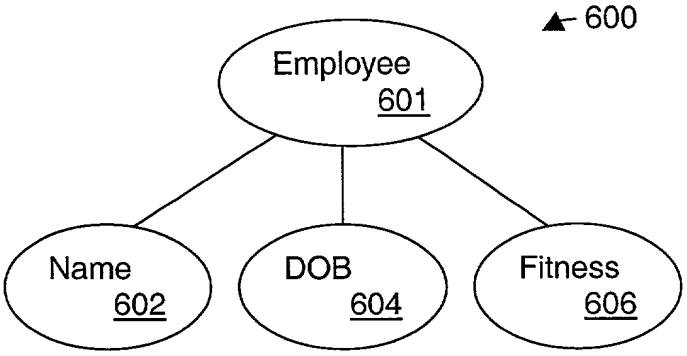

FIG. 6A (Prior Art)

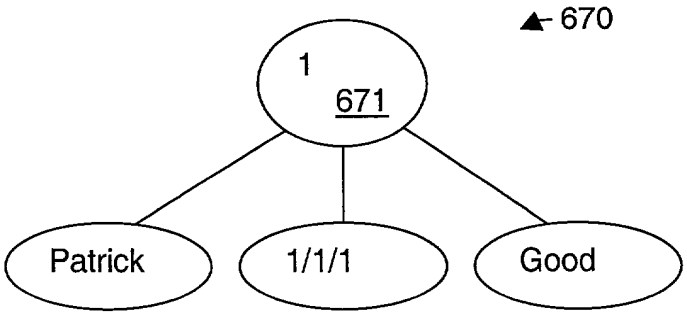

FIG. 6B (Prior Art)

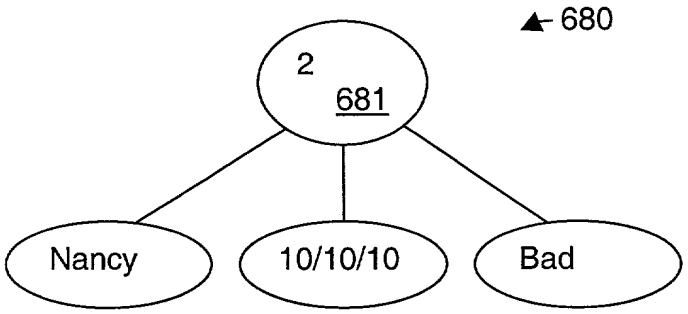

FIG. 6C (Prior Art)

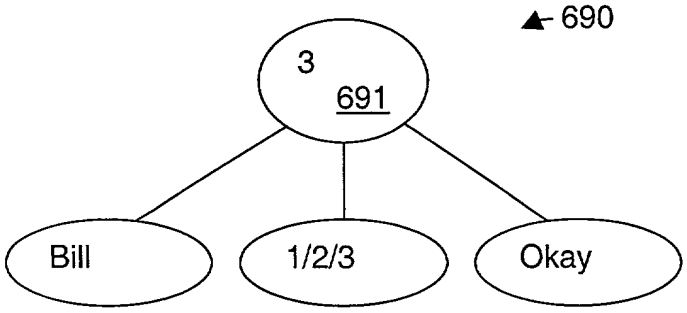

FIG. 6D (Prior Art)

EDS2356.2-DIV

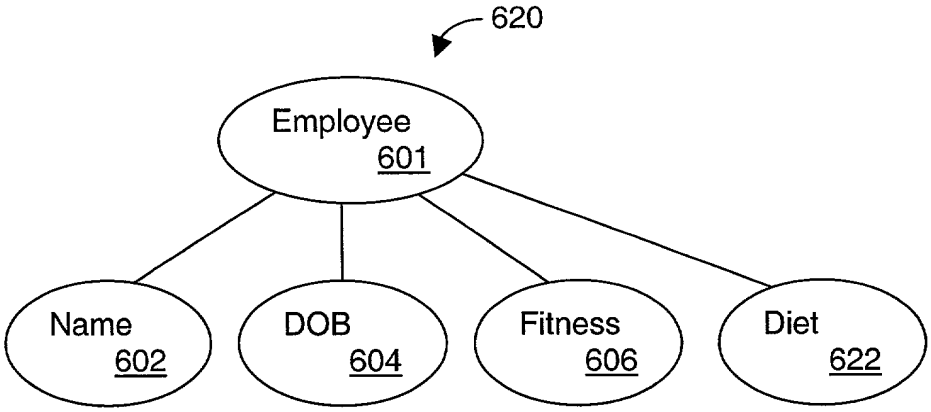

FIG. 6E (Prior Art)

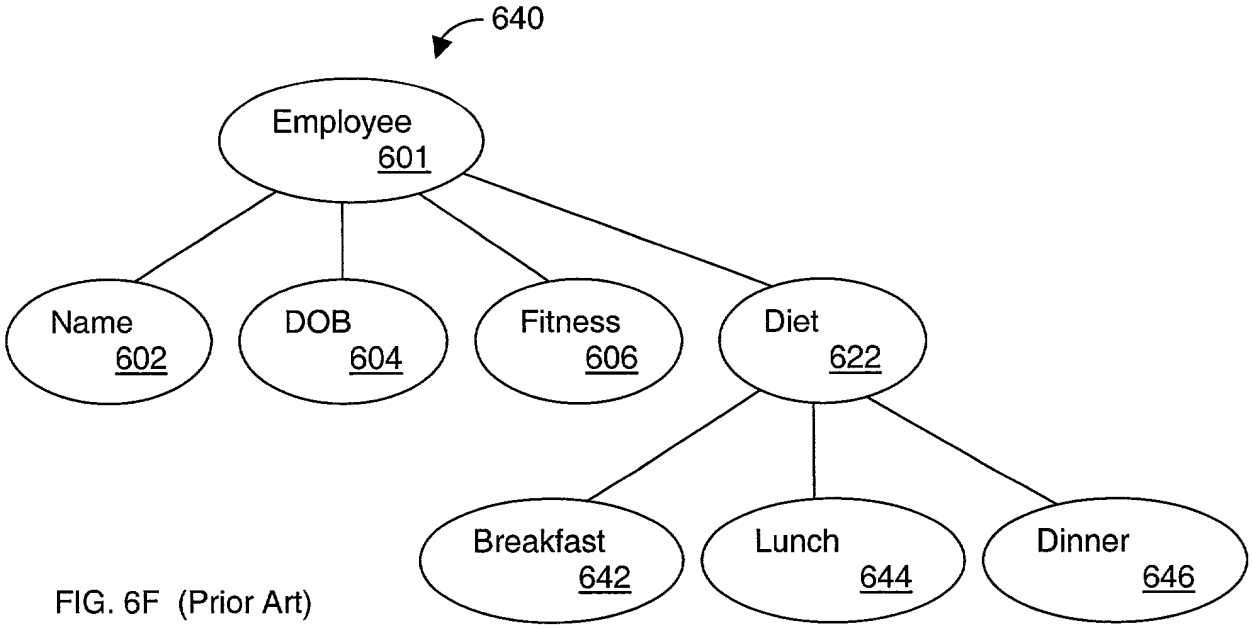

FIG. 6F (Prior Art)

Docket No. 2005-0426

700

```
701   <Employee  type = "Manager">
702         <Name> Patrick </Name>
703         <DOB> 1907 </DOB>
704         <Fitness/>
705         <Diet>
706               <Breakfast> Doughnuts </Breakfast>
707               <Lunch> Pizza </Lunch>
708               <Dinner> Beer </Dinner>
709         </Diet>
710   </Employee>
```

(File: Patrick.xml)

FIG. 7

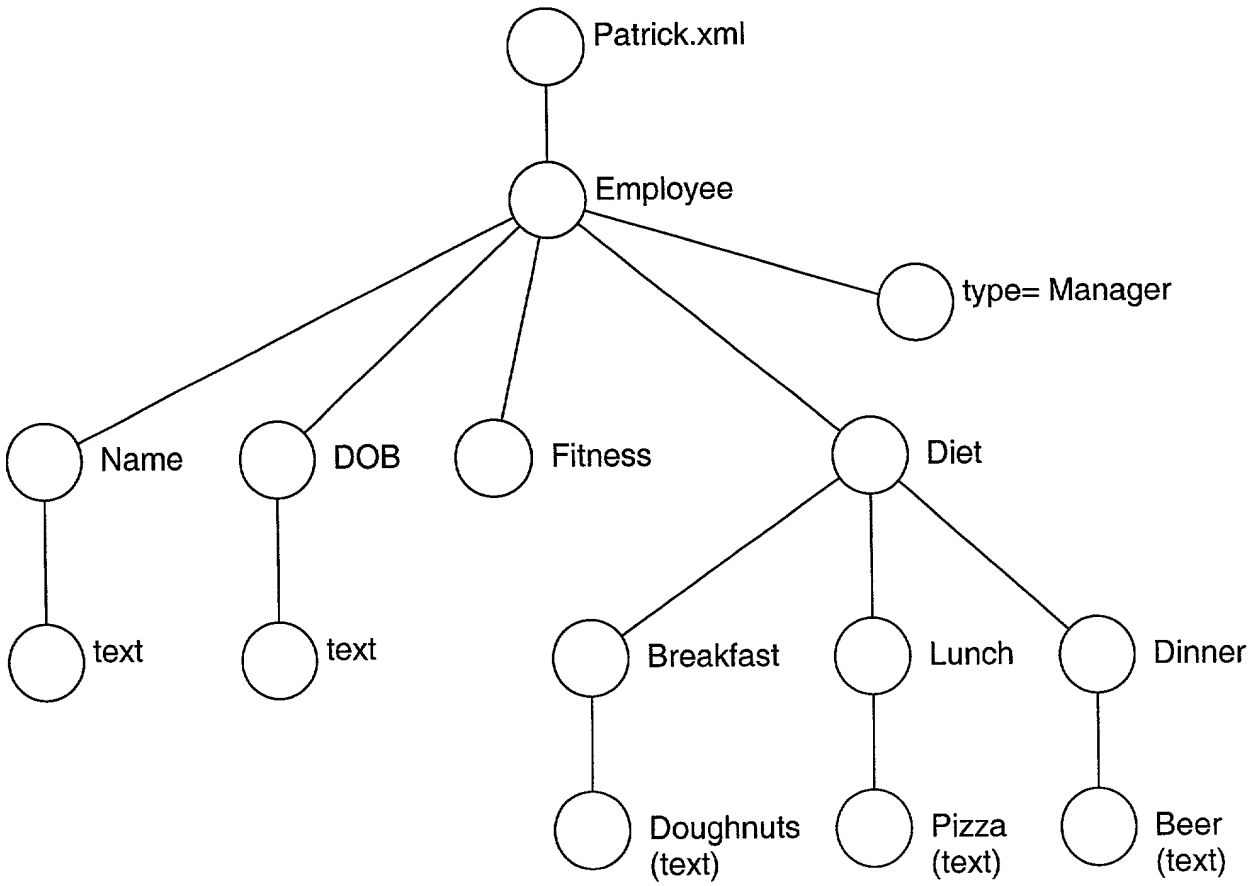

FIG. 8

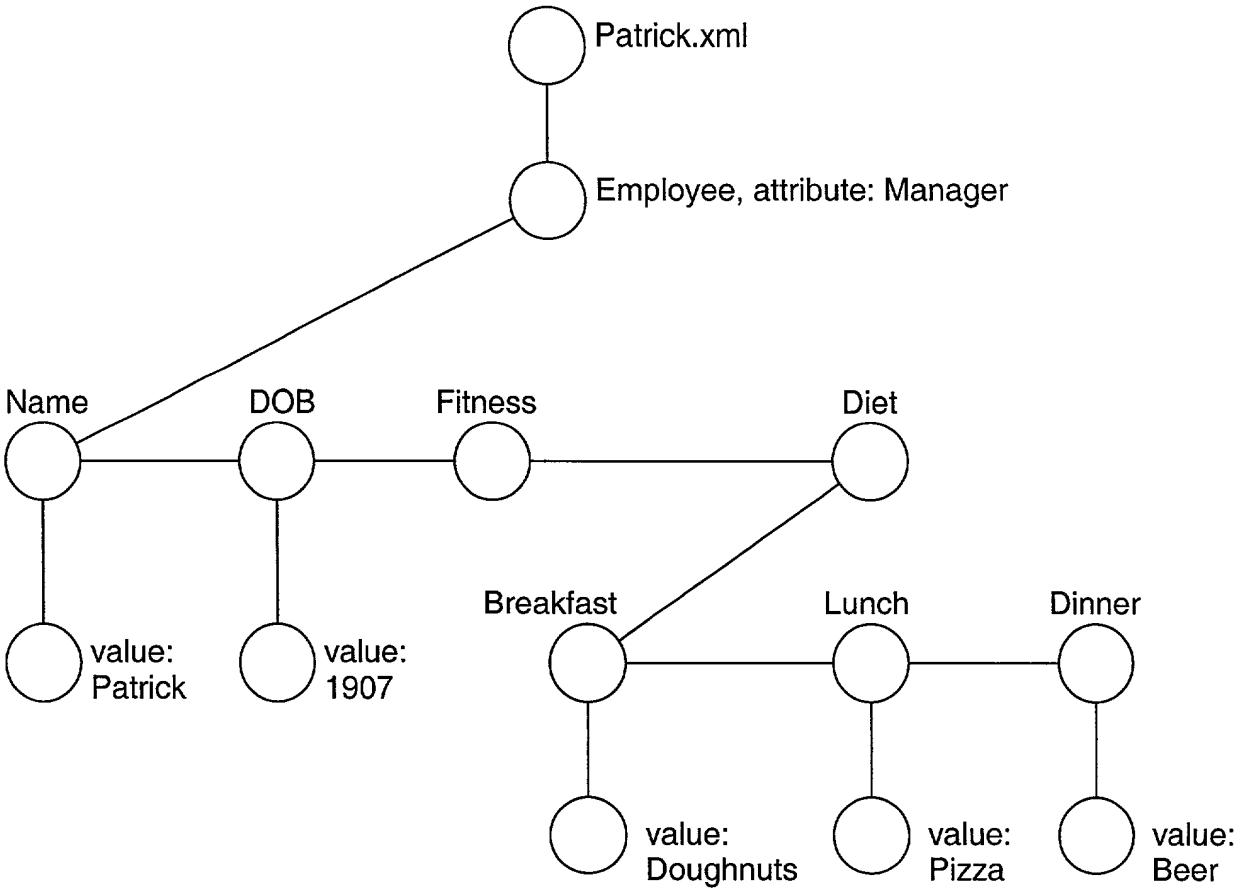

FIG. 9

Patrick - Exhibit 1005 - Page 8

1000

	Node ID	attribute	tag value	parent ID	first child ID	next sibling ID
1001	Patrick.xml				Employee	
1002	Employee	Manager		Patrick.xml	Name	
1003	Name		Patrick	Employee		DOB
1004	DOB		1907			Fitness
1005	Fitness					Diet
1006	Diet				Breakfast	
1007	Lunch		Pizza			Dinner
1008	Dinner		Beer			
1009	Breakfast		Doughnuts	Diet		Lunch

FIG. 10

Patent Application Publication Oct. 29, 2009

1100

T1	Node ID	attribute	tag value	parent ID	child array
1101	Patrick.xml				T2
1102	Employee	Manager		Patrick.xml	T2
1103	Name		Patrick	Employee	
1104	DOB		1907	Employee	
1105	Fitness			Employee	
1106	Diet			Employee	T2
1107	Lunch		Pizza	Diet	
1108	Dinner		Beer	Diet	
1109	Breakfast		Doughnuts	Diet	

FIG. 11A

1180

T2	Node ID	child ID
1181	Patrick.xml	Employee
1182	Employee	Name
1183	Employee	Diet
1184	Employee	DOB
1185	Employee	Fitness
1186	Diet	Breakfast
1187	Diet	Dinner
1188	Diet	Lunch

FIG. 11B

1190

	Node ID	child ID
1191	Patrick.xml	Employee
1192	Employee	Name, Diet, DOB, Fitness
1193	Diet	Breakfast, Dinner, Lunch

FIG. 11C

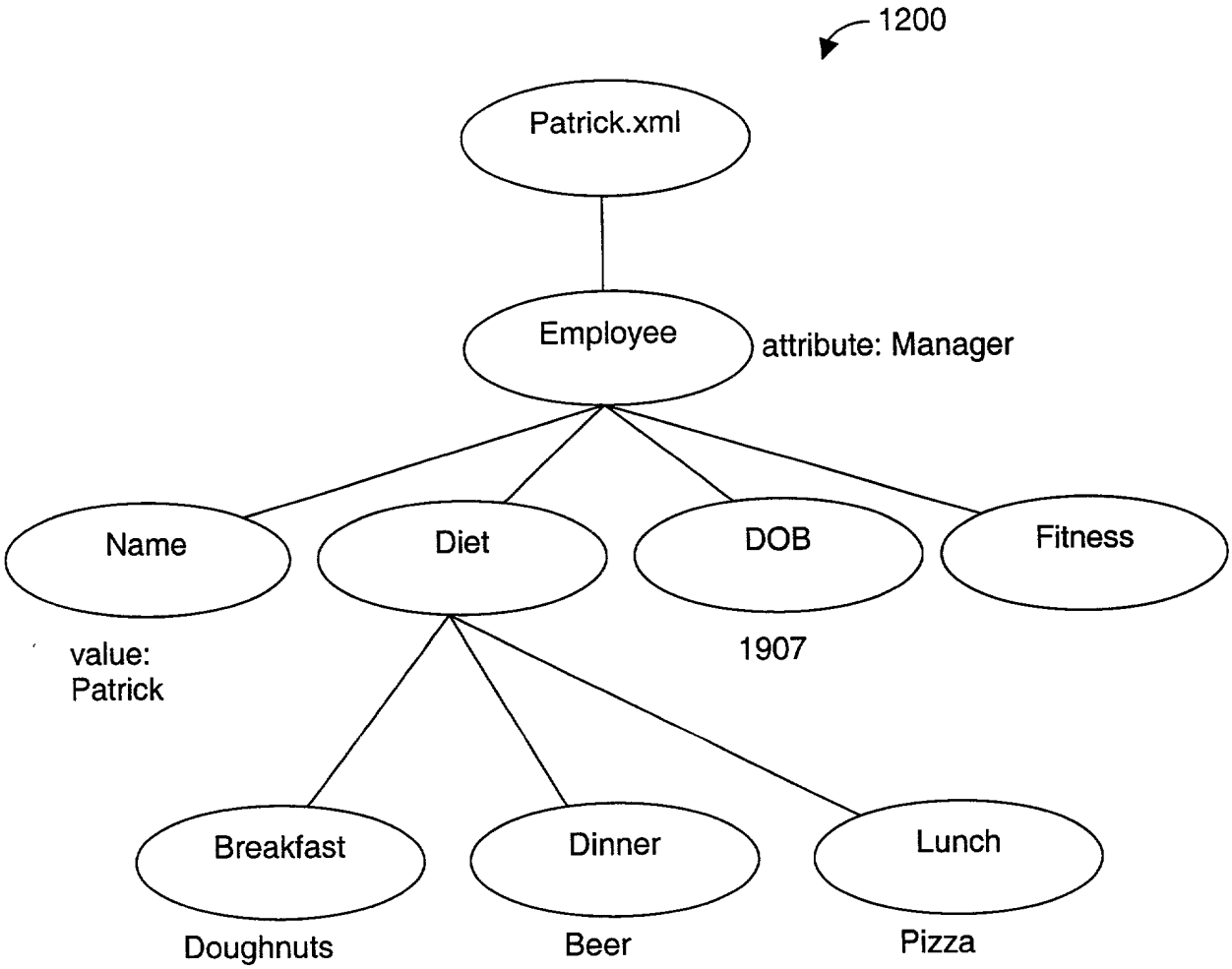
1200
Patrick.xml
Employee
attribute: Manager
Name
Diet
DOB
Fitness
value:
Patrick
1907
Breakfast
Dinner
Lunch
Doughnuts
Beer
Pizza

FIG. 12

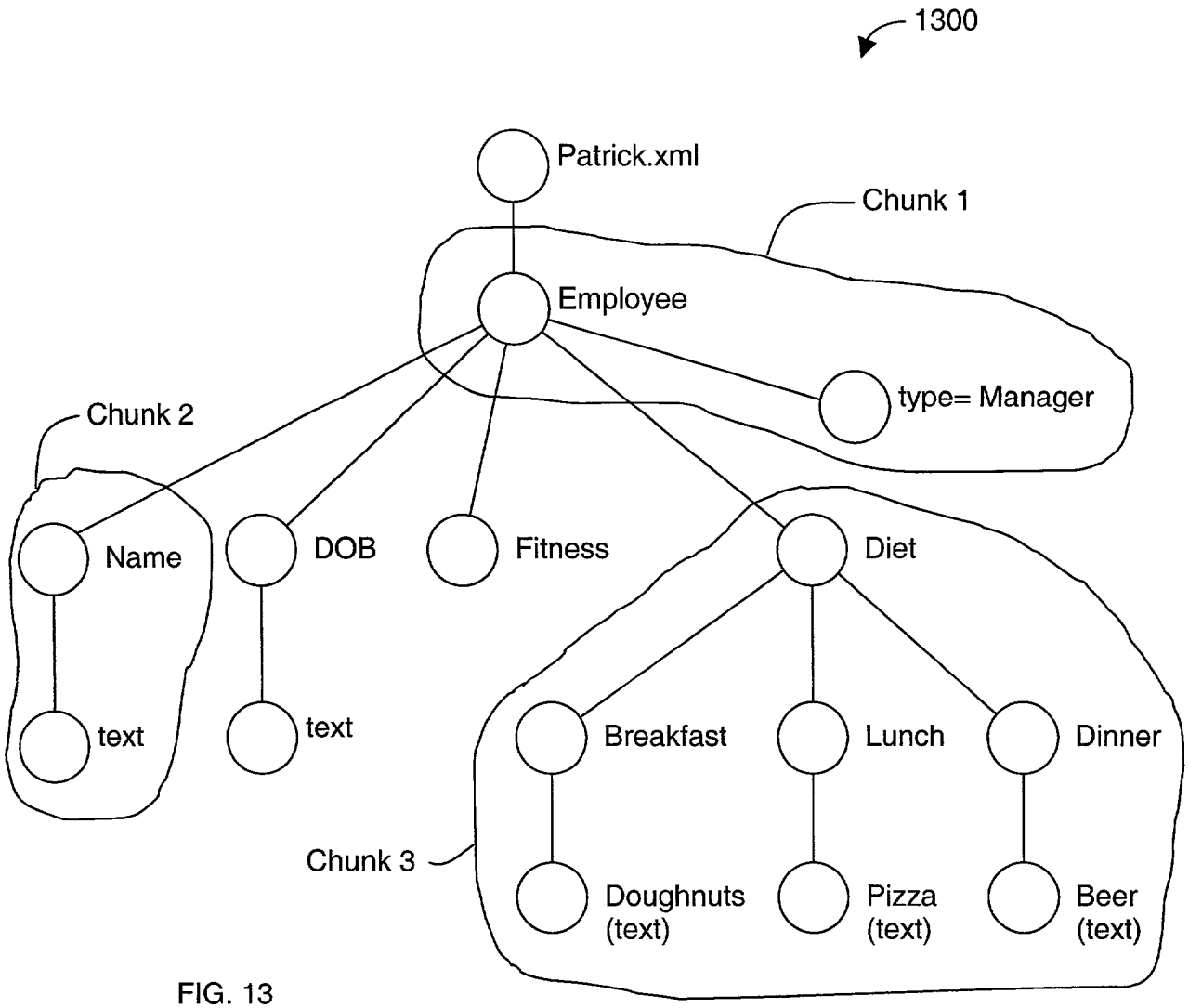

FIG. 13

Patent Application Publication

T1	Node ID	attribute	tag value	parent	child
1401	Patrick.xml				Chunk 1
1402	Chunk 1			Patrick.xml	T2
1403	DOB		1907	Chunk 1	
1404	Fitness			Chunk 1	

FIG. 14A

T2	Node ID	child ID
1481	Chunk 1	Chunk 2
1482	Chunk 1	DOB
1483	Chunk 1	Chunk 3
1484	Chunk 1	Fitness
1485	Diet	Breakfast
1486	Diet	Dinner
1487	Diet	Lunch

FIG. 14B

Chunk 1	Node ID	attribute
1421	Employee	Manager

FIG. 14C

Chunk 2	Node ID	value
1431	Name	Patrick

FIG. 14D

Chunk 3	Node ID	attribute	tag value	parent	child
1441	Diet			Chunk 1	T2
1442	Lunch		Pizza	Diet	
1443	Dinner		Beer	Diet	
1444	Breakfast		Doughnuts	Diet	

FIG. 14E

1500

object	field	key	type	nulls	comment
document	document_id	pk	number	no	
	url		string	no	
	text_content		clob		supports limited text search
	binary_content		blob		doesn't need to be re-parsed

FIG. 15

1600

object	field	key	type	nulls	comment
document	document_id	pk	number	no	
	url		string	no	
node	document_id	fk; pk1:1	number	no	cascading delete
	preorder_position	pk1:2	number	no	
	owner_position		number		parent's position
	class_id	fk; ie	number	no	
	search_path	ie	string	no	e.g. "ns1:tag1/ns2:tag2/..."
	value		string		
	blob_value		blob		
	child_document_id	fk	number		used when linking to a child document
class	class_id	pk	number	no	
	node_type		number	no	Node.ELEMENT_NODE, etc.
	local_name	ie; ie1:2	string		
	namespace_id	fk; ie1:1	number		
	prefix		string		
namespace	namespace_id	pk	number	no	
	namespace_uri	ak	string	no	

FIG. 16

APPLE-1005 Page 15

1700

object	field	key	type	nulls	comment
node	node_id	pk	number	no	
	owner_edge_id	fk	number	no	cascading delete
	class_id	fk; ie	number	no	
	first_attribute_edge_id	fk	number		
	first_child_edge_id	fk	number		
	value		string		
	blob_value		blob		
edge	edge_id	pk	number	no	
	parent_node_id	fk	number	no	cascading delete
	child_node_id	fk; ie	number	no	
	next_edge_id	fk	number		
	previous_edge_id	fk	number	no	
class	class_id	pk	number	no	
	node_type		number	no	Node.ELEMENT_NODE, etc.
	local_name	ie; ie1:2	string		
	namespace_id	fk; ie1:1	number		
	prefix		string		
namespace	namespace_id	pk	number	no	
	namespace_uri	ak	string	no	

FIG. 17

Page 16 of 18

1800

object	field	key	type	nulls	comment
attribute_type	edge_id		number	no	globally unique id
	class_id	fk	number	no	
	value		string		
child_type	edge_id		number	no	globally unique id
	node_id	fk	number	no	
	link_id	fk	number		
node	node_id	pk	number	no	
	owner_node_id	fk	number	no	cascading delete
	owner_edge_id		number	no	
	class_id	fk; ie	number	no	
	attributes		array		array of attribute_type objects
	children		array		array of child_type objects
	value		string		
	blob_value		blob		
link	edge_id	pk	number	no	
	parent_node_id	fk	number		cascading delete
	child_node_id	fk; ie	number		
class	class_id	pk	number	no	
	node_type		number	no	Node.ELEMENT_NODE, etc.
	local_name	ie; ie1:2	string		
	namespace_id	fk; ie1:1	number		
	prefix		string		
namespace	namespace_id	pk	number	no	
	namespace_uri	ak	string	no	

FIG. 18

Patent Application Publication

1900

object	field	key	type	nulls	comment
member_type	edge_id		number	no	globally unique id
	class_id	fk; ie	number		
	attribute_count		number		
	child_count		number		
	value		string		
	child_chunk_id	fk	number		used when linking to a child chunk
chunk	chunk_id	pk	number	no	
	owner_chunk_id	fk	number	no	cascading delete
	owner_edge_id		number	no	
	root_class_id	fk; ie	number	no	
	members		array		array of member_type objects
	blob_value		blob		
link	edge_id	pk	number	no	
	parent_chunk_id	fk	number		cascading delete
	child_chunk_id	fk; ie	number		
class	class_id	pk	number	no	
	node_type		number	no	Node.ELEMENT_NODE, etc.
	local_name	ie; ie1:2	string		
	namespace_id	fk; ie1:1	number		
	prefix		string		
namespace	namespace_id	pk	number	no	
	namespace_uri	ak	string	no	

FIG. 19

Sheet 18 of 18